Neural Net studies

1 Summary of previous results

Previous results are documented in the note that can be found in the Higgs-Dilepton group meeting agenda for July 15. Summary of previous results:

- Detailed cell-level info may provide extra S/B discrimination (in addition to cluster variables). This needs to be studied.
- NN based on the HMatrix variables does not perform better S/B-wise than HMatrix does. Needs to be investigated.

2 What can we expect from NN

Why do we expect NN to perform better than current EMID ? For two reasons (one of the two or both):

- NN can deal with nonlinear relationships between input variables. On the other hand, if all input variables are correlated/anticorrelated with each other (concentric elliptical 2D distributions for pairs of input variables) then no difference should be expected between HMatrix and NN based on the same variables. (The requirement that the input variables are Gaussian distributed is a stronger one, provided that the variables are correlated, this requirement is an overconstraint as far as improving the S/B discrimination with NN. This requirement ensures that the χ^2-like function that we use for EMID would follow true χ^2 distribution but it is not necessary for the S/B discrimination improvement)
- Due to user-friendly NN implementation in ROOT (much easier and faster to use than HMatrix machinery) we could hope to find better (more S/B discriminating) variables than the HMatrix uses to train the NN.

3 Outline of this update

1. Sanity checks of NN training and performance.
2. 2D distributions of input variables.
3. Incorporating cell info variables into NN.

Currently we limit our studies to CC and use the samples described in the previous note.

4 Sanity checks of NN training and performance.

The following sanity checks have been performed:

- Change BFGS learing method parameters (RESET and TAU).
- Train and test NN on a set of orthogonal samples.

- Reduce number of variables (from 7 to 4).
- Scale input variables (multiply each of them by 2).
- Change the order of input variables.

4.1 Change BFGS learing method parameters (RESET and TAU).

User-tunable parameters of the BFGS learning method are RESET(search direction is reset to steepest descent every RESET epochs) and TAU (lower TAU = higher precision, slower search). Default values are RESET=50, and TAU=3.0. Figure 1 shows NN training curve and NN output for the modified values of RESET and TAU while Figure 2 shows the NN training curve and NN output for the default values of RESET and TAU parameters.

4.2 Train and test NN on a set of orthogonal samples.

We split the input sample previously used in two so that 4 orthogonal samples are available (denote them A,B,C,D). "Sample" here means a combination of signal sample and background sample, e.g. sample A = A(signal) + A(background) Then the following configurations are used:

- A – training, B – testing.
- B – training, A – testing.
- C – training, D – testing.
- C – training, C – testing.

The results are shown in Figure 3

4.3 Reducing the number of variables.

We reduce the number of variables from 7 to 4 and compare NN training curves and NN output for 7-14-1 and 4-8-1 configurations (Figure 4).

4.4 Scale input variables (multiply each of them by 2)

We scale each input variable (multiply it by 2) and make sure that the values of NN output are not sensitive to such scaling (Figure 5).

4.5 Change the order of input variables.

We make sure that NN output is not sensitive to the order in which input variables are used. Figure 6 shows NN training curve and NN output for two different ordering of 7 NN input variables.

5 2D distributions of input variables.

We look at 2D distributions of (some of) the input variables with the emphasis on the most discriminating ones, i.e. EM3 rϕ-width and track isolation. Figures 7, 9, and 8 show some of these distributions for signal and background. Are these distributions sufficently non-elliptical and non-concentric to expect any help from NN ? Is the requirement that "the variables are correlated/anticorrelated" weaker than the requirement that "the 2D distributions are elliptical and concentric" or the same one phrased differently ?

6 Incorporating cell info variables into NN.

Our next step is to incorporate cell-level variables (see previous update) into the NN net. We choose best S/B-discriminating cell-level variables and combine them with some of the cluster variables. We end up with the 10-variable NN that uses the following input variables:

1. EM3 rϕ-width,
2. Number of cells in EM1 above 100 MeV in the cone shell of 0.1<dR(η, ϕ)<0.7,
3. Number of cells in EM2 above 100 MeV in the cone shell of 0.1<dR(η, ϕ)<0.7,
4. Number of cells in EM3 above 100 MeV in the cone shell of 0.1<dR(η, ϕ)<0.7,
5. Number of cells in EM4 above 100 MeV in the cone shell of 0.1<dR(η, ϕ)<0.7,
6. Track isolation,
7. log(E),
8. sum of EM1 floor cell energies(threshold = 100 MeV) in the cone shell of 0.1<dR(η, ϕ)<0.7,
9. ratio of sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.1 to the sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.7,
10. ratio of sum of cell energies in EM4 floor (threshold = 100 MeV) within a conic shell of 0.1<dR(η, ϕ)<0.7 to the sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.7.

We use 15 input nodes (the program complains when the number goes up to 16) and 1500 epochs for training with BFGS method using default parameter values.

7 List of what has NOT been done / or not with much care.

1. η-binning was not done. The whole η range of CC is used for training.
2. Didn't optimize "Number of input variables vs Number of events in the training sample". What is optimal ?

3. Didn't try MC for training.

4. Interpretation of NN performance is based on one (Efficiency, fake rate) point, which is most suitable for comparison with "HMatrix + Track Isolation" EMID performance. Didn't do the "Efficiency vs. fake rate curve". This comparison was done for the reference NN (based on 7 variables, BFGS method with default parameter settings, training sample of 2500 signal + 5000 background events and orthogonal test sample of 2500 signal + 5000 background events). For all other NN's used we compare visually the NN output plots. In principle, this approach does not give a full info about comparison of NN "HMatrix + Track Isolation" EMID. At the present stage – while our goal is to achive significant background suppression – this approach is sufficient and could be refined later with "Efficiency vs. fake rate curve".

8 Current conclusions, questions, and plans

- NN passed all of the consistency checks ? What do 2 peaks in some of the NN output distributions mean ?
- Are 2D distributions sufficently non-linear to expect help from NN ?
- Incoroporating best cell-info variables doesn't seem to improve S/B...
- something optimistic should be put in this bullet for balance.

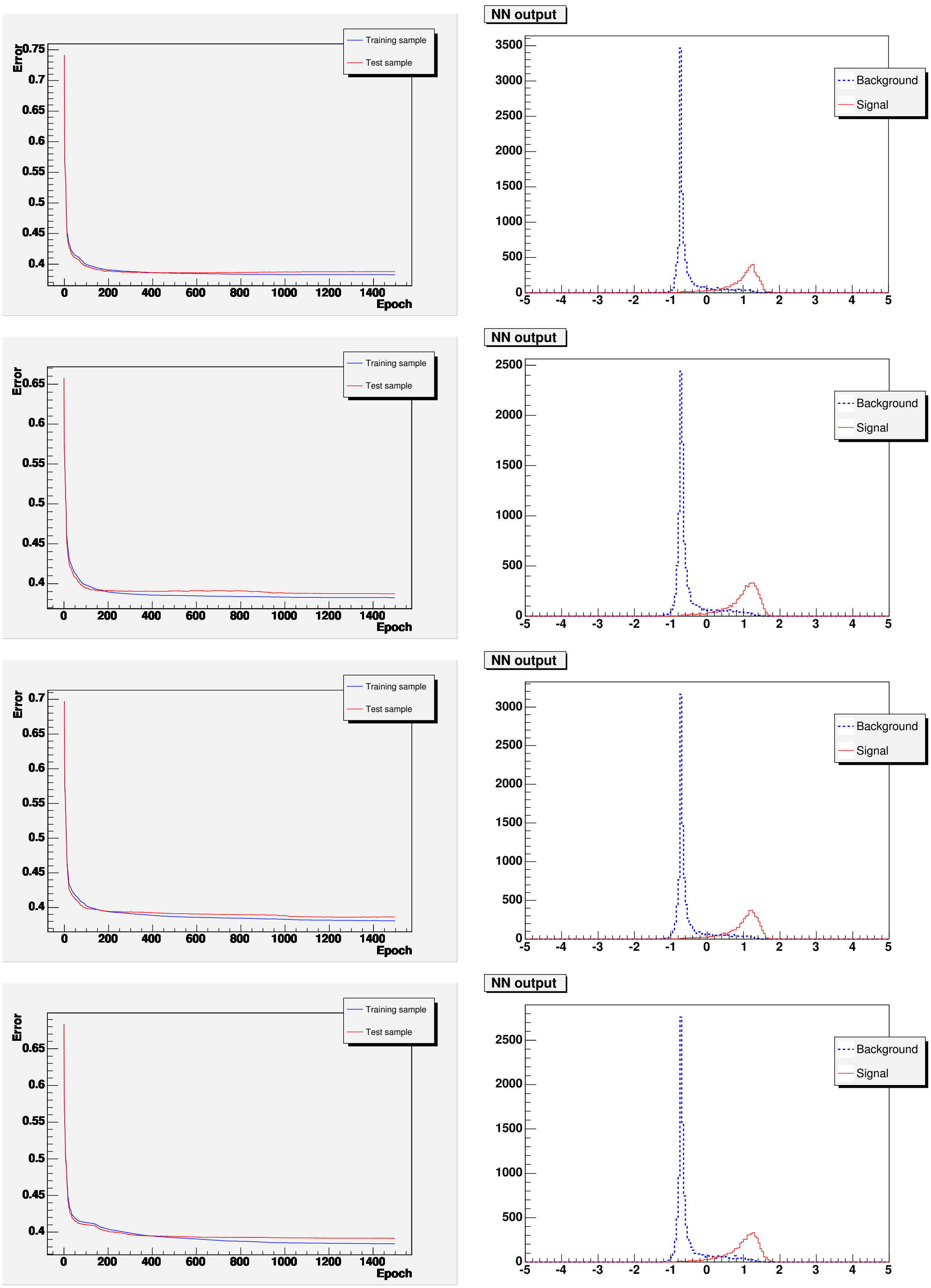

Figure 1: Testing NN performance by modifying BFGS learing method parameters RESET and TAU. NN training curves and NN output in CC for BFGS learing method with modified BFGS method parameters. Top: RESET=50, TAU=2.0. Top Middle: RESET=50, TAU=4.0. Bottom Middle: RESET=25, TAU=4.0. Bottom: RESET=10, TAU=5.0. Left: error for training and test sample as a function of the number of epochs (total number of training epochs for each set of parameters is 1500). Right: NN output. 7 NN variables are used: those in HMx7 minus Z(primary vertex) (EM3 rϕ-width, four

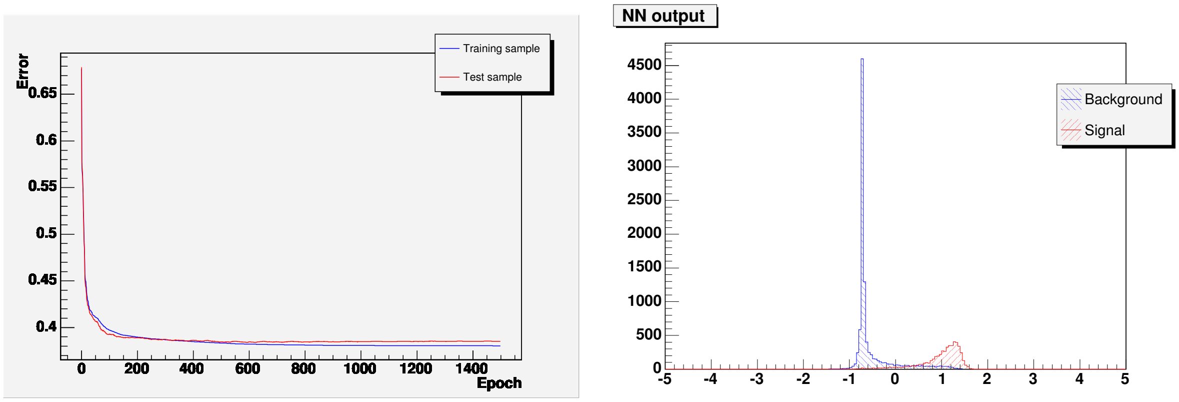

Figure 2: NN training curves and NN output in CC for BFGS learing method. Default values of BFGS method parameters are used: RESET=50, TAU=3.0. Left: error for training and test sample as a function of the number of epochs (total number of training epochs is 1500). Right: NN output. 7 NN variables are used: those in HMx7 minus Z(primary vertex) (EM3 rϕ-width, four EM floor energy fractions, log(E)) plus track isolation. One hidden layer of 14 hidden nodes is used.

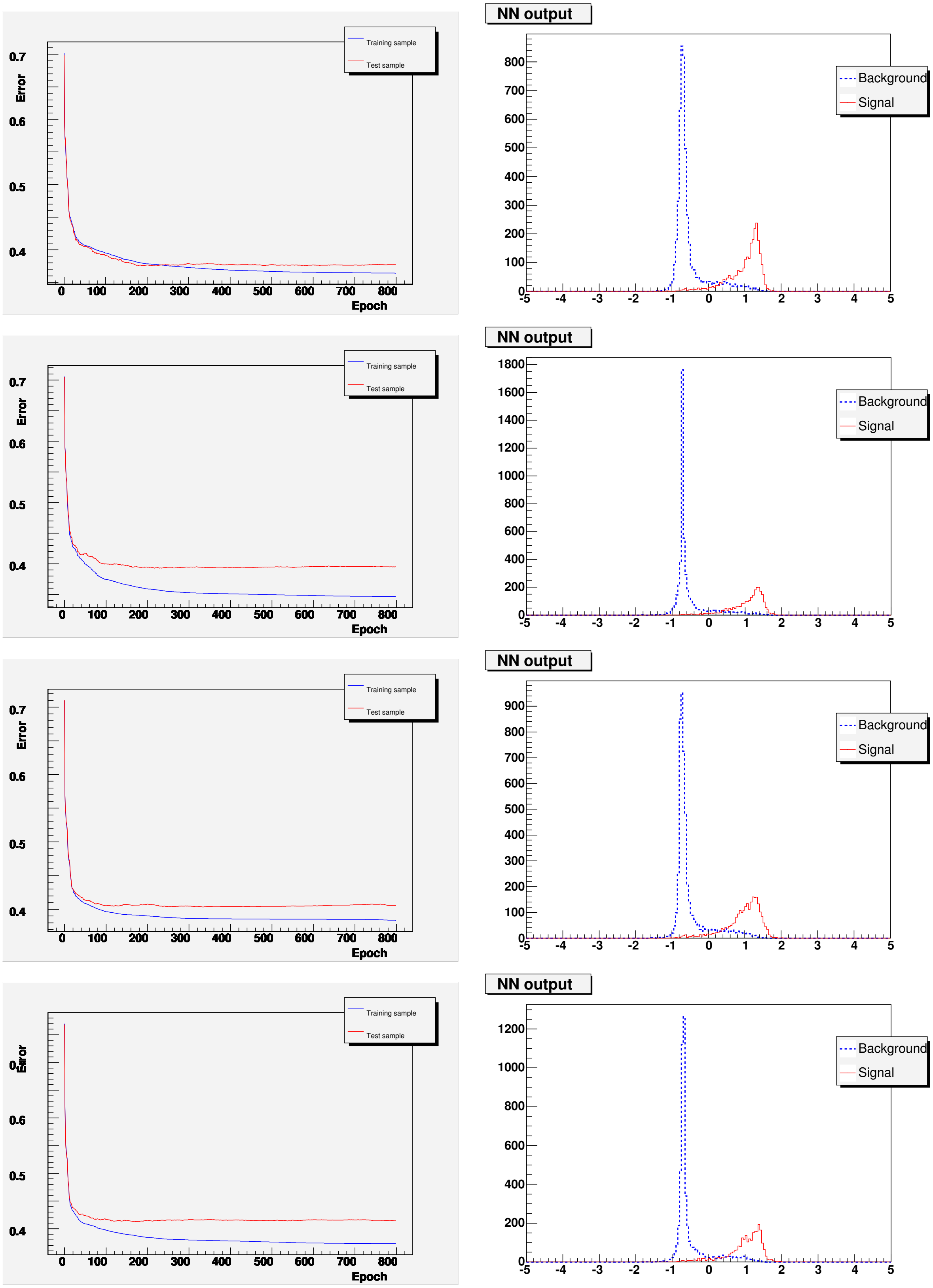

Figure 3: Testing NN performance on a set of orthogonal samples. NN training curves and NN output in CC for BFGS learing method 4 different sets of training/testing samples. We use 4 orthogonal samples A,B,C,D each of which is contains 1300 signal entries and 2500 background entries. Top: A – training, B – testing. Top Middle: B – training, A – testing. Bottom Middle: C – training, D – testing. Bottom: D – training, C – testing. Left: error for training and test sample as a function of the number of epochs (total number of training epochs for each set of samples is 800). Right: NN output. 7

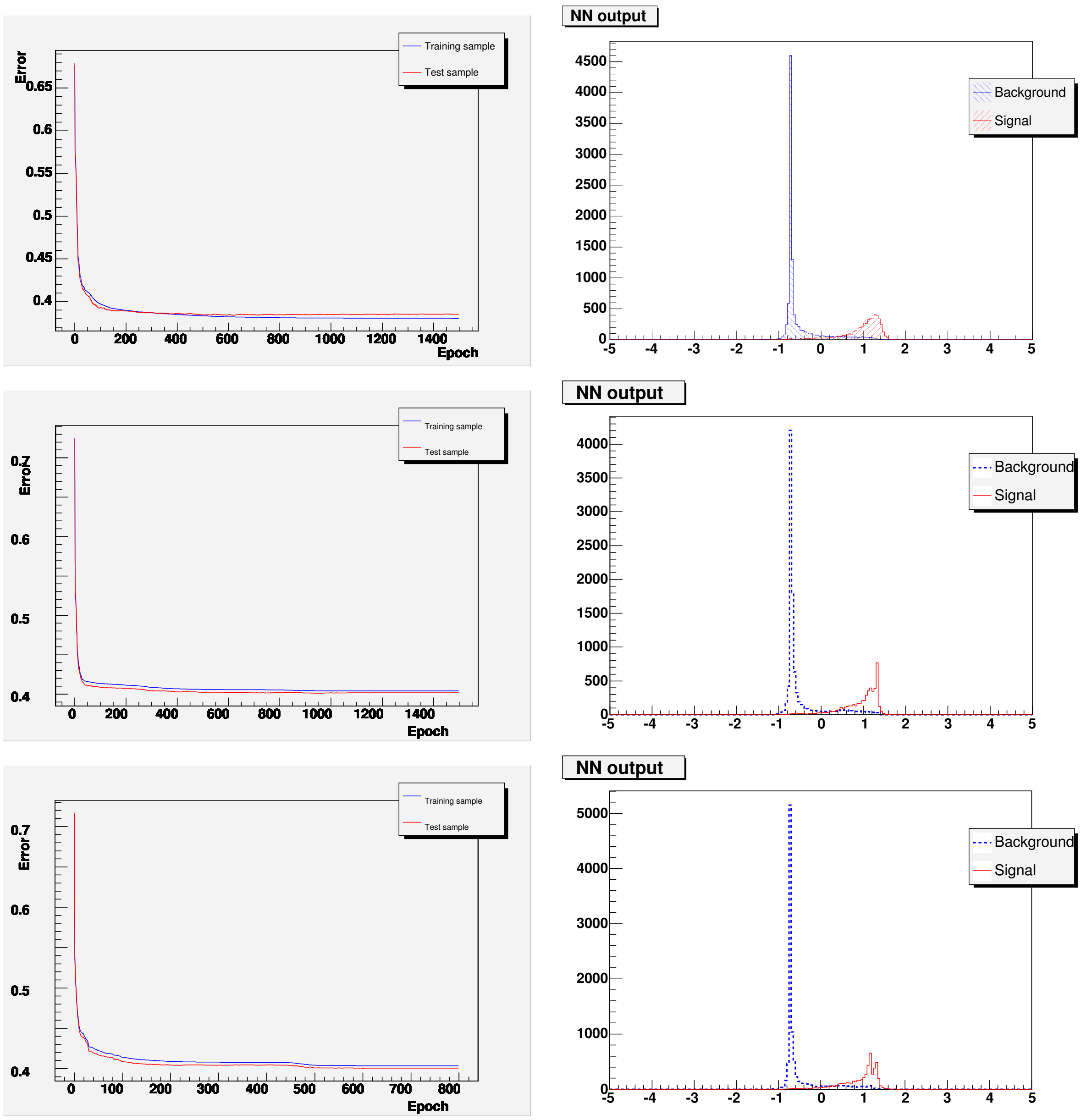

Figure 4: Testing NN performance by reducing the number of variables. Left: error for training and test sample as a function of the number of epochs. Right: NN output. Top – 7 NN variables are used: those in HMx7 minus Z(primary vertex) (EM3 rϕ-width, four EM floor energy fractions, log(E)) plus track isolation. One hidden layer of 14 hidden nodes is used (number of training epochs is 1500) Middle – 4 NN variables are used: EM3 rϕ-width, isolation, track isolation, log(E), i.e. 7 variables in the top row minus 4 EM floor energy fractions plus isolation. One hidden layer of 8 hidden nodes is used (number of training epochs is 1500) Bottom – same as middle except that 800 training epochs are used instead of 1500.

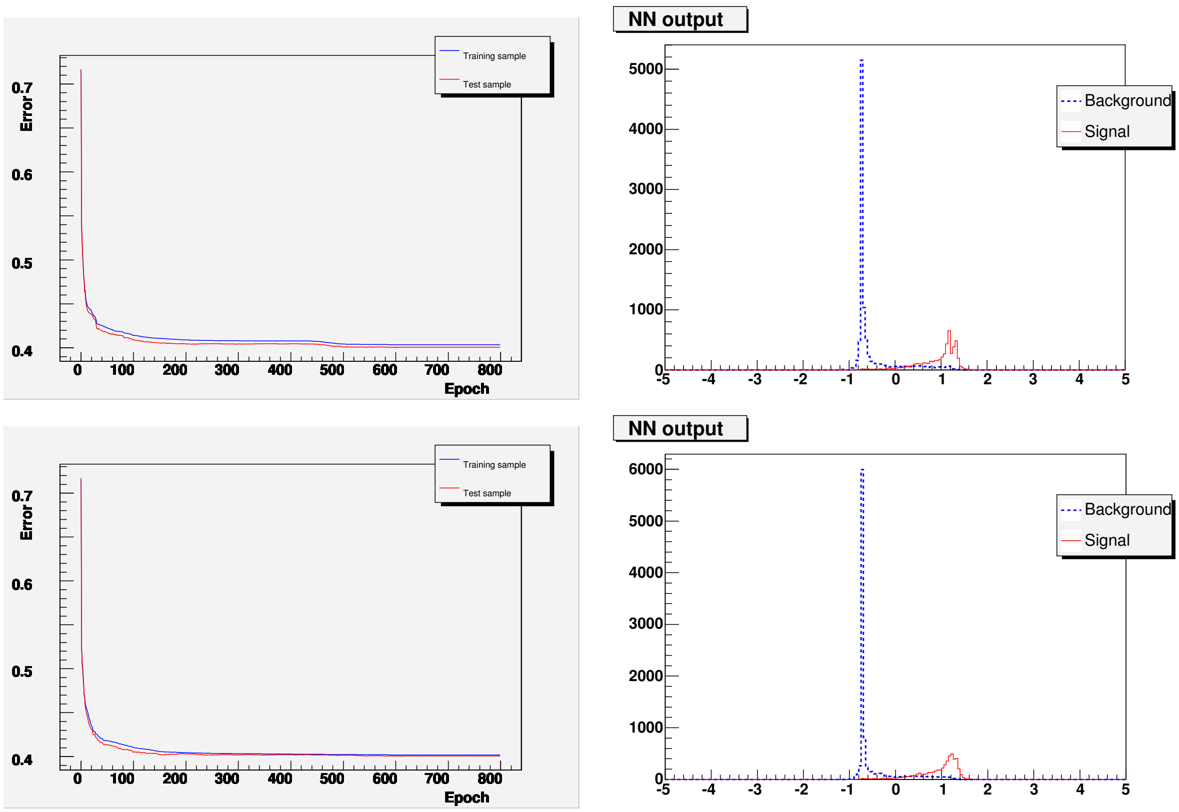

Figure 5: Testing NN robustness by scaling input variables (multiplying each of the input variables by 2). Left: error for training and test sample as a function of the number of epochs. Right: NN output. We use 4 NN variables are used: EM3 rϕ-width, isolation, track isolation, log(E). One hidden layer of 8 hidden nodes is used (number of training epochs is 800) Top – no scaling is applied. Bottom – each input variable is multiplied by 2.

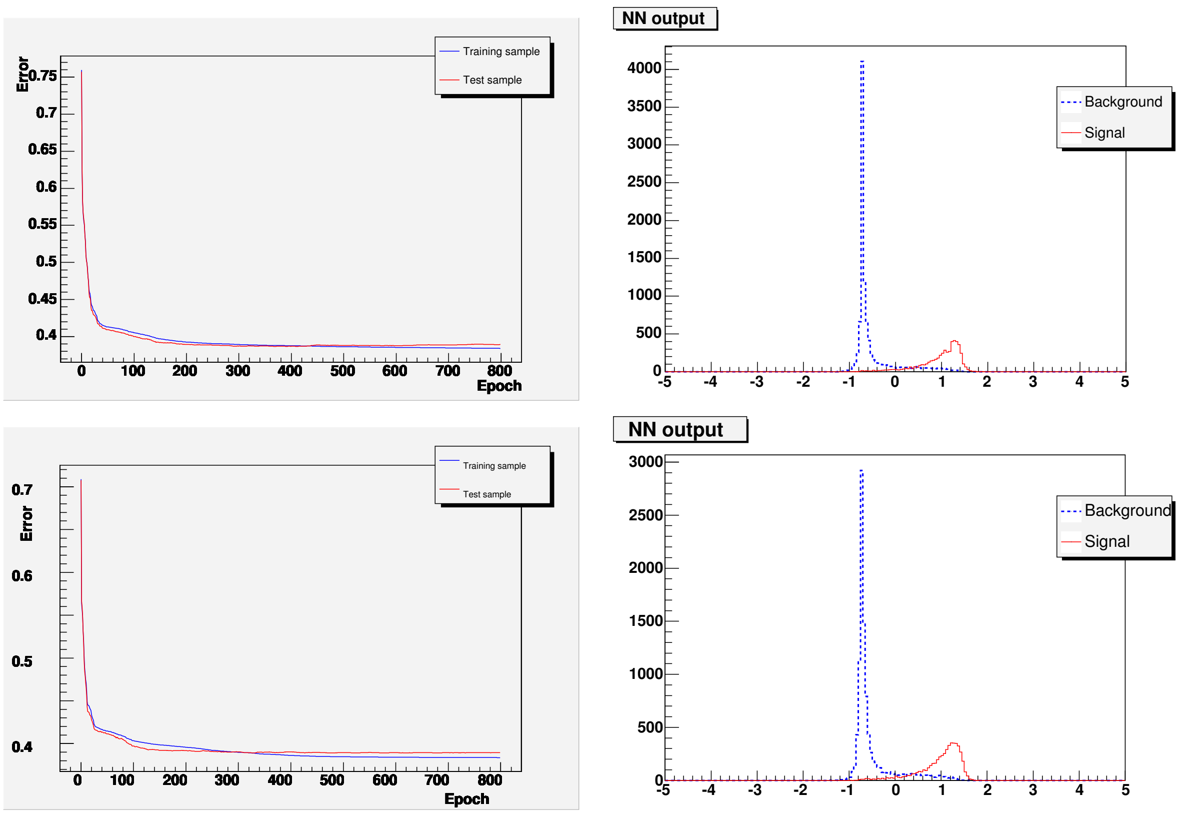

Figure 6: Testing NN robustness by changing the order of input variables. We use 7 NN variables : those in HMx7 minus Z(primary vertex) (EM3 rϕ-width, four EM floor energy fractions, log(E)) plus track isolation. One hidden layer of 14 hidden nodes is used (number of training epochs is 800) Left: error for training and test sample as a function of the number of epochs. Right: NN output. Top – the input variables are used in the following order: EM3 rϕ-width, EM1 fraction, EM2 fraction, EM3 fraction, EM4 fraction, track isolation, log(E). Bottom – the input variables are used in the following order: log(E), EM1 fraction, track isolation, EM2 fraction, EM3 rϕ-width, EM3 fraction, EM4 fraction.

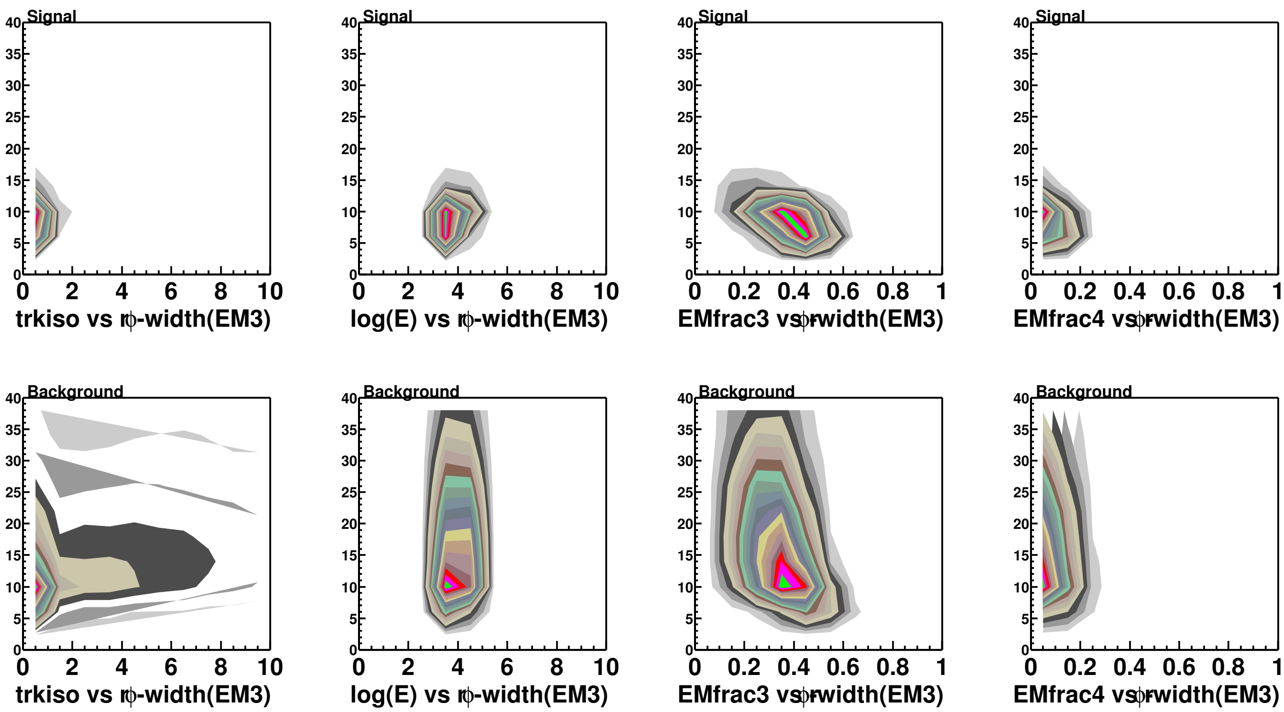

Figure 7: 2D distributions of "ala HMatrix-7" NN input variables (set 1). Left: Track isolation vs. EM3 rϕ-width. Middle Left: log(E) vs. EM3 rϕ-width. Middle Right: EM3 energy fraction vs. EM3 rϕ-width. Right: EM4 energy fraction vs. EM3 rϕ-width. Top: signal. Bottom: background.

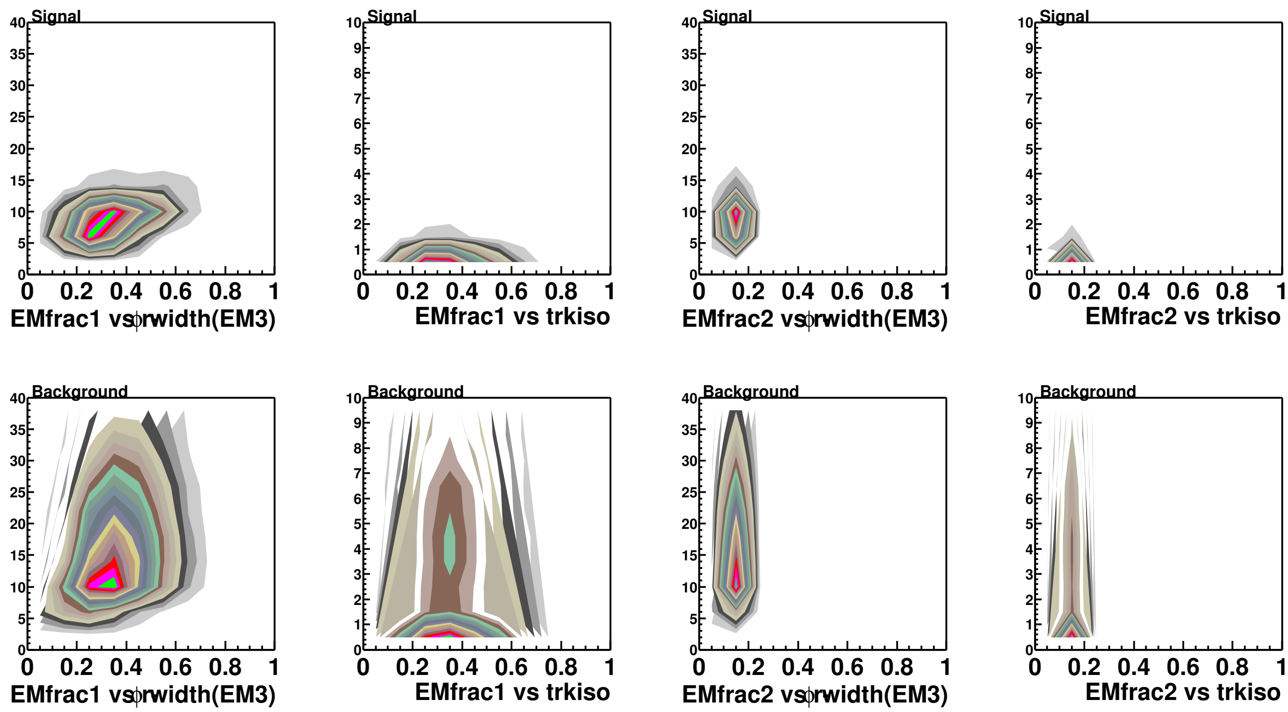

Figure 8: 2D distributions of "ala HMatrix-7" NN input variables (set 2). Left: EM1 energy fraction vs. EM3 rϕ-width. Middle Left: EM1 energy fraction vs. track isolation. Middle Right: EM2 energy fraction vs. EM3 rϕ-width. Right: EM2 energy fraction vs. track isolation. Top: signal. Bottom: background.

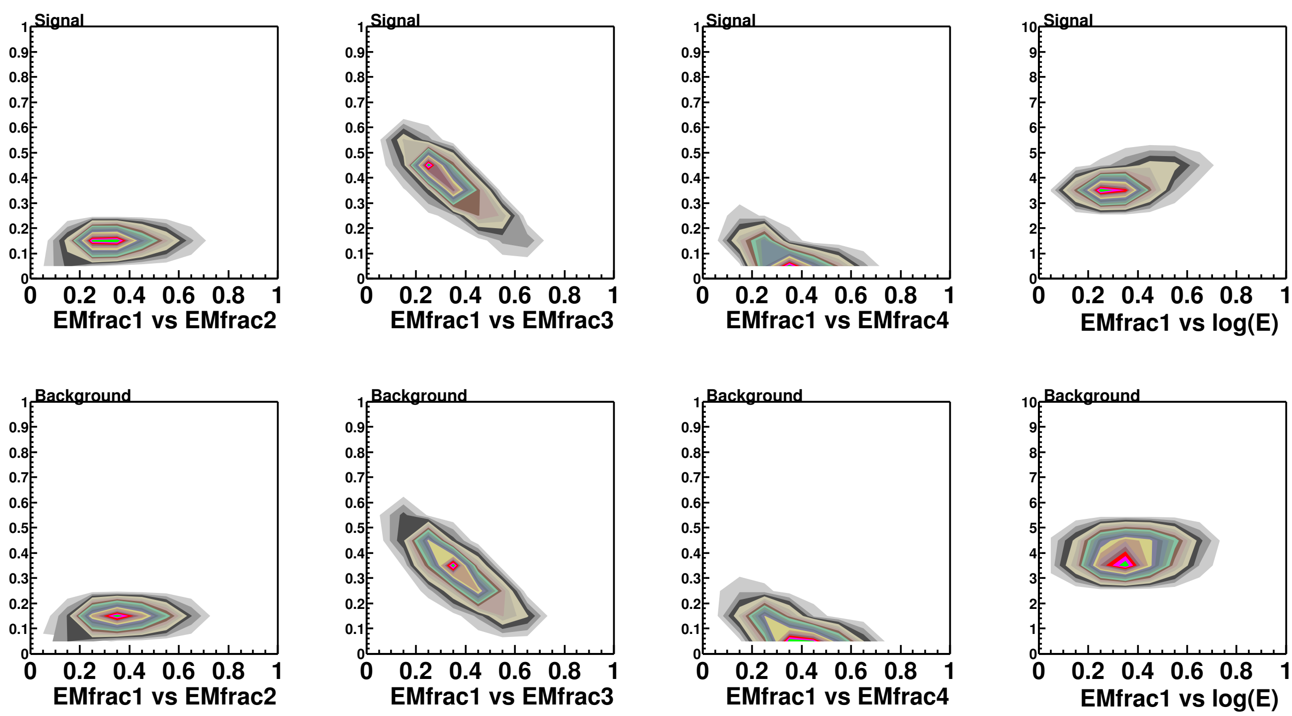

Figure 9: 2D distributions of "ala HMatrix-7" NN input variables (set 3). Left: EM1 energy fraction vs. EM2 energy fraction. Middle Left: EM1 energy fraction vs. EM3 energy fraction. Middle Right: EM1 energy fraction vs. EM4 energy fraction. Right: EM1 energy fraction vs. log(E). Top: signal. Bottom: background.

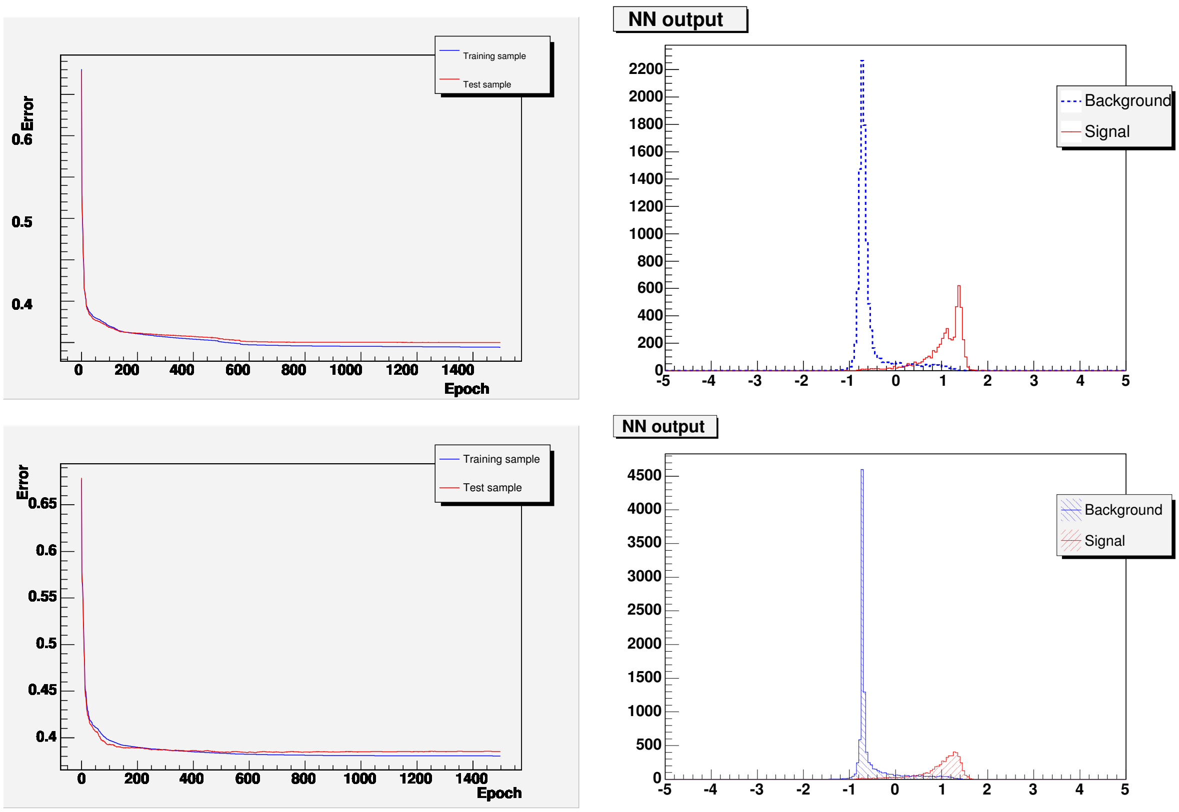

Figure 10: NN training curves and NN output in CC for BFGS learing method for the NN to which the cell-info variables are added(10 variables total) and "ala HMatrix-7" 7 variable NN. Default values of BFGS method parameters are used: RESET=50, TAU=3.0, number of training epochs is 1500. Top: 10 NN variables are used: EM3 rϕ-width, Number of cells above 100 MeV in the cone shell of 0.1<dR(η, ϕ)<0.7 for each of the four EM floors, track isolation, log(E), sum of EM1 floor cell energies(threshold = 100 MeV) in the cone shell of 0.1<dR(η, ϕ)<0.7, ratio of sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.1 to the sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.7, ratio of sum of cell energies in EM4 floor (threshold = 100 MeV) within a conic shell of 0.1<dR(η, ϕ)<0.7 to the sum of cell energies in EM4 floor (threshold = 100 MeV) within dR(η, ϕ)<0.7. One hidden layer of 15 hidden nodes is used. Bottom: 7 NN variables are used: those in HMx7 minus Z(primary vertex) (EM3 rϕ-width, four EM floor energy fractions, log(E)) plus track isolation. One hidden layer of 14 hidden nodes is used Left: error for training and test sample as a function of the number of epochs. Right: NN output.